STAT 705 Generalized additive models

Timothy Hanson

Department of Statistics, University of South Carolina

Stat 705: Data Analysis II

Generalized additive models

Consider a linear regression problem:

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i,$$

where $e_1, \ldots, e_n \overset{iid}{\sim} N(0, \sigma^2)$.

* Diagnostics (residual plots, added variable plots) might indicate poor fit of the basic model above.
* Remedial measures might include transforming the response, transforming one or both predictors, or both.
* One also might consider adding quadratic terms and/or an interaction term.
* Note: we only consider transforming *continuous* predictors!

When considering a transformation of one predictor, an added variable plot can suggest a transformation (e.g. $\log(x), 1/x$) that might work *if the other predictor is "correctly" specified*.

In general, a transformation is given by a function $x^* = g(x)$. Say we decide that x_{i1} should be log-transformed and the reciprocal of x_{i2} should be used. Then the resulting model is

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 \log(x_{i1}) + \beta_2 / x_{i2} + \epsilon_i \\ &= \beta_0 + g_{\beta_1}(x_{i1}) + g_{\beta_2}(x_{i2}) + \epsilon_i, \end{aligned}$$

where $g_{\beta_1}(x)$ and $g_{\beta_2}(x)$ are two functions specified by β_1 and β_2.

Here we are specifying forms for $g_1(x|\beta_1)$ and $g_2(x|\beta_2)$ based on exploratory data analysis, but we could from the outset specify *models* for $g_1(x|\boldsymbol{\theta}_1)$ and $g_2(x|\boldsymbol{\theta}_2)$ that are rich enough to capture interesting and predictively useful aspects of how the predictors affect the response and *estimate these functions from the data*.

One example of this is through a basis expansion; for the jth predictor the transformation is:

$$g_j(x) = \sum_{k=1}^{K_j} \theta_{jk}\psi_{jk}(x),$$

where $\{\psi_{jk}(\cdot)\}_{k=1}^{K_j}$ are B-spline basis functions, or sines/cosines, etc. This approach has gained more favor from Bayesians, but is not the approach taken in SAS PROC GAM. PROC GAM makes use of *cubic smoothing splines*.

This is an example of "nonparametric regression," which ironically connotes the inclusion of *lots* of parameters rather than fewer.

Smoothing spline

For simple bivariate regression data $\{(x_i, y_i)\}_{i=1}^n$, a cubic spline smoother $g(x)$ minimizes

$$\sum_{i=1}^{n}(y_i - g(x_i))^2 + \lambda \int_{-\infty}^{\infty} g''(x)^2 dx.$$

Good fit is achieved by minimizing the sum of squares $\sum_{i=1}^n (y_i - g(x_i))^2$. The $\int_{-\infty}^{\infty} g''(x)^2 dx$ term measures how wiggly $g(x)$ is and $\lambda \geq 0$ is how much we will penalize $g(x)$ for being wiggly.

A spline trades off between goodness of fit and wiggliness.

Although not obvious, the solution to this minimization is a cubic spline: a piecewise cubic polynomial with the pieces joined at the unique x_i values.

Choosing λ

Hastie and Tibshirani (1986, 1990) point out that the meaning of λ depends on the units x_i is measured in, but that λ can be picked to yield an "effective degrees of freedom" *df* or an "effective number of parameters" being used in $g(x)$. Then the complexity of $g(x)$ is equivalent to $(df - 1)$-degree polynomial, but with the coefficients "spread out" more yielding a more flexible function that fits data better.

Alternatively, λ can be picked through cross validation, by minimizing

$$CV(\lambda) = \sum_{i=1}^{n} (y_i - g_\lambda^{-i}(x_i))^2.$$

Both options are available in SAS.

Generalized additive model

We have $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_1, \ldots, y_n$ are normal, Bernoulli, or Poisson. The generalized additive model (GAM) is given by

$$h\{E(Y_i)\} = \beta_0 + g_1(x_{i1}) + \cdots + g_k(x_{ik}),$$

for p predictor variables. Y_i is a member of an exponential family such as binomial, Poisson, normal, etc. h is a link function.

Each of $g_1(x), \ldots, g_p(x)$ are modeled via cubic smoothing splines, each with their own smoothness parameters $\lambda_1, \ldots, \lambda_p$ either specified as $df_1, \ldots, df_p$ or estimated through cross-validation. The model is fit through "backfitting." See Hastie and Tibshirani (1990) or the SAS documentation for details.

PROC GAM in SAS

SAS actually fits the model

$$h\{E(Y_i)\} = \beta_0 + \underbrace{\beta_1 x_{i1} + \tilde{g}_1(x_{i1})}_{g_1(x_{i1})} + \cdots + \underbrace{\beta_k x_{ik} + \tilde{g}_k(x_{ik})}_{g_k(x_{ik})},$$

where $\tilde{g}_1(\cdot), \dots, \tilde{g}_k(\cdot)$ have the linear part detrended.

SAS provides plots of the $\tilde{g}_1(\cdot), \dots, \tilde{g}_k(\cdot)$ as well as tests $H_0 : \tilde{g}_j(\cdot) = 0$ for $j = 1, \dots, k$. These tests are that a transformation is *not required* for each variable, i.e. linear is okay.

Unfortunately, if we reject $H_0 : \tilde{g}_j(\cdot) = 0$, the best plot to look at is of $g_j(x) = \beta_j x + \tilde{g}_j(x)$ versus x spanning the range of $x_{1j}, \dots, x_{nj}$. This is provided in the R package GAM (also in DPpcakge for R, a Bayesian version), but not from SAS.

Satellite counts Y_i

Let's fit a GAM to the crab mating data:

```
proc gam plots(unpack)=components(clm) data=crabs;
 class spine color;
 model satell=param(color) spline(weight)  / dist=poisson;
run;
```

This fits the model

$$Y_i \sim \text{Pois}(\mu_i),$$

$$\log(\mu_i) = \beta_0 + \beta_1 I\{c_i = 1\} + \beta_2 I\{c_i = 2\} + \beta_3 I\{c_i = 3\} + \beta_4 \text{wt}_i + g_4(\text{wt}_i).$$

SAS output

Output:

```
                    The GAM Procedure
               Dependent Variable: satell
         Regression Model Component(s): color
     Smoothing Model Component(s): spline(weight)

               Summary of Input Data Set

   Number of Observations                    173
   Number of Missing Observations              0
   Distribution                          Poisson
   Link Function                             Log

                Class Level Information
       Class       Levels    Values
       color            4    1, 2, 3, 4

             Iteration Summary and Fit Statistics

Number of local scoring iterations                      6
Local scoring convergence criterion          3.011103E-11
Final Number of Backfitting Iterations                  1
Final Backfitting Criterion                  2.286359E-10
The Deviance of the Final Estimate           532.81821791
```

SAS parameter estimates

Regression Model Analysis
Parameter Estimates

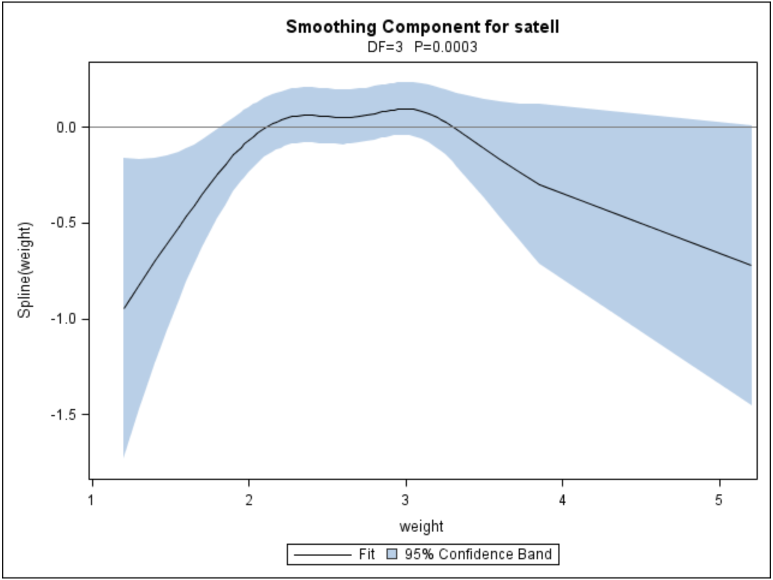
Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	-0.50255	0.23759	-2.12	0.0359
color 1	0.36148	0.20850	1.73	0.0848
color 2	0.21891	0.16261	1.35	0.1801
color 3	-0.01158	0.18063	-0.06	0.9490
color 4	0	.	.	.
Linear(weight)	0.56218	0.07894	7.12	<.0001

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(weight)	3.00000	18.986722	18.9867	0.0003

The Analysis of Deviance table gives a χ^2-test from comparing the deviance between the full model and the model with this variable dropped: here the model with color (categorical) *plus only a linear effect in weight*. We see that weight is significantly nonlinear at the 5% level. The default $df = 3$ corresponds to a smoothing spline with the complexity of a cubic polynomial.

The following plot has the estimated smoothing spline function with the linear effect subtracted out. The plot includes a 95% confidence band for the whole curve. We visually inspect where this band does not include zero to get an idea of where significant nonlinearity occurs. This plot can suggest simpler transformations of predictor variables than use of the full-blown smoothing spline: here maybe a quadratic?


Smoothing Component for satell
DF=3 P=0.0003
Spline(weight)
0.0
-0.5
-1.0
-1.5
1
2
3
4
5
weight
Fit
95% Confidence Band

The band shows a pronounced deviation from linearity for weight. The plot spans the range of weight values in the data set and becomes highly variable at the ends. Do you think extrapolation is a good idea using GAMs?

Note: You *can* get predicted values out of SAS with CIs. Just stick to representative values.

PROC TRANSREG for normal data

PROC GAM handles Poisson, Bernoulli, normal, and gamma data. If you only have normal data, PROC TRANSREG will fit a very general transformation model, for example

$$h(Y_i) = \beta_0 + g_1(x_{i1}) + g_2(x_{i2}) + \epsilon_i,$$

and provide estimates of $h(\cdot)$, $g_1(\cdot)$, and $g_2(\cdot)$.

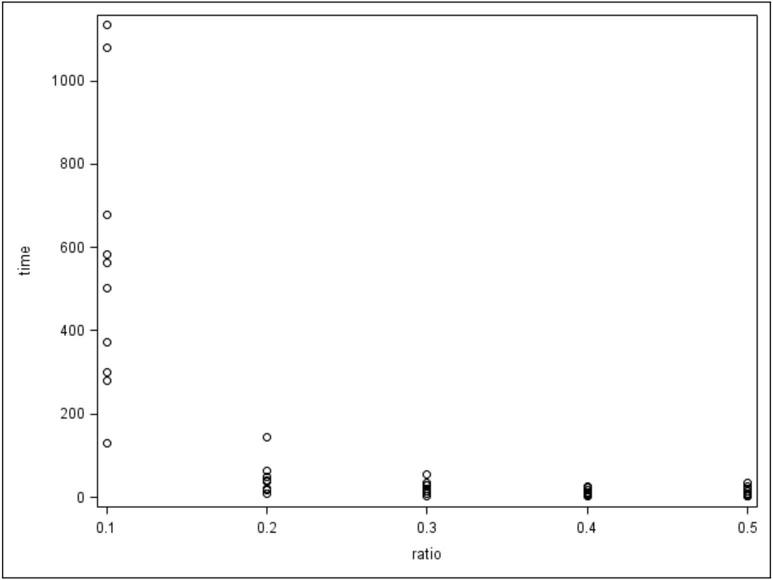
$h(\cdot)$ can simply be the Box-Cox family, indexed by λ, or a very general spline function.

Electrical components

* Consider time-to-failure in minutes of $n = 50$ electrical components.
* Each component was manufactured using a ratio of two types of materials; this ratio was fixed at 0.1, 0.2, 0.3, 0.4, and 0.5.
* Ten components were observed to fail at each of these manufacturing ratios in a designed experiment.
* It is of interest to model the failure-time as a function of the ratio, to determine if a significant relationship exists, and if so to describe the relationship simply.

SAS code: data & plot

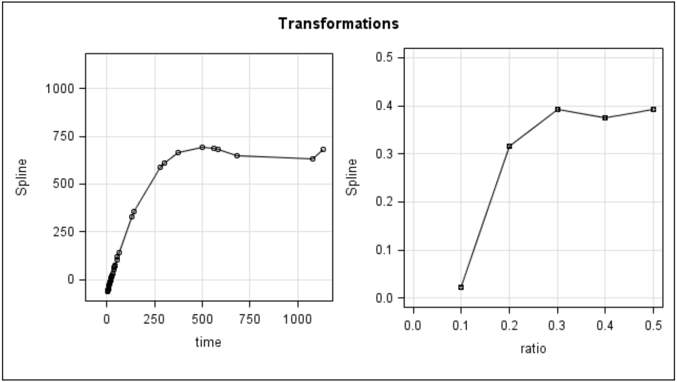
```
data elec;
input ratio time @@;
datalines;
0.5     34.9 0.5     9.3 0.5     6.0 0.5     3.4 0.5    14.9
0.5      9.0 0.5    19.9 0.5     2.3 0.5     4.1 0.5    25.0
0.4     16.9 0.4    11.3 0.4    25.4 0.4    10.7 0.4    24.1
0.4      3.7 0.4     7.2 0.4    18.9 0.4     2.2 0.4     8.4
0.3     54.7 0.3    13.4 0.3    29.3 0.3    28.9 0.3    21.1
0.3     35.5 0.3    15.0 0.3     4.6 0.3    15.1 0.3     8.7
0.2      9.3 0.2    37.6 0.2    21.0 0.2   143.5 0.2    21.8
0.2     50.5 0.2    40.4 0.2    63.1 0.2    41.1 0.2    16.5
0.1    373.0 0.1   584.0 0.1  1080.1 0.1   300.8 0.1   130.8
0.1    280.2 0.1   679.2 0.1   501.6 0.1  1134.3 0.1   562.6
;
ods pdf; ods graphics on;
proc sgscatter; plot time*ratio; run;
ods graphics off; ods pdf close;
```

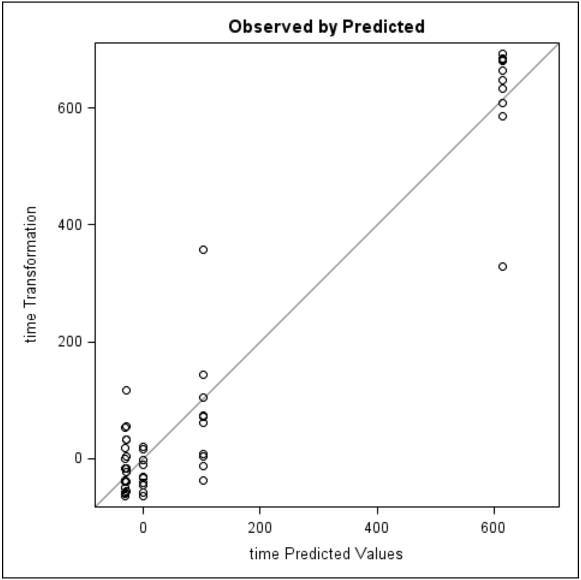


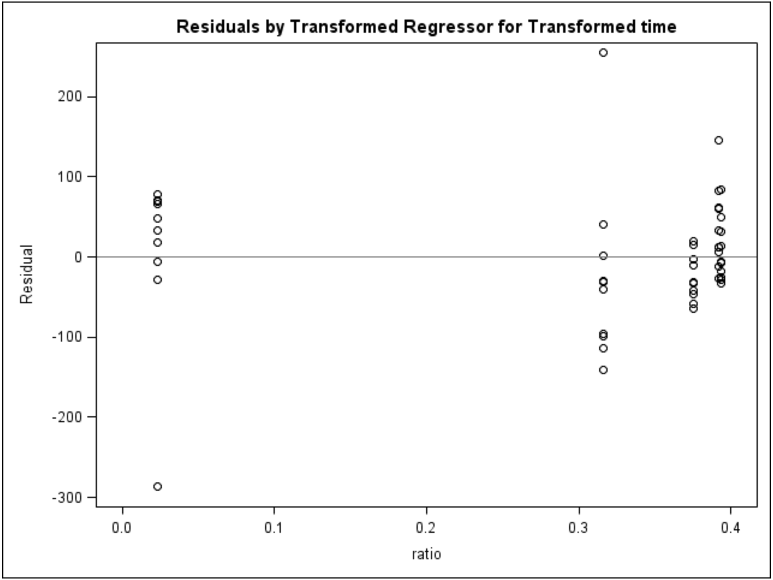
1000
800
600
400
200
0
time
0.1
0.2
0.3
0.4
0.5
ratio

SAS code: fit $h(Y_i) = \beta_0 + g_1(x_{i1}) + \epsilon_i$

```
ods pdf; ods graphics on;
proc transreg data=elec solve ss2 plots=(transformation obp residuals);
 model spline(time) = spline(ratio); run;
ods graphics off; ods pdf close;
```

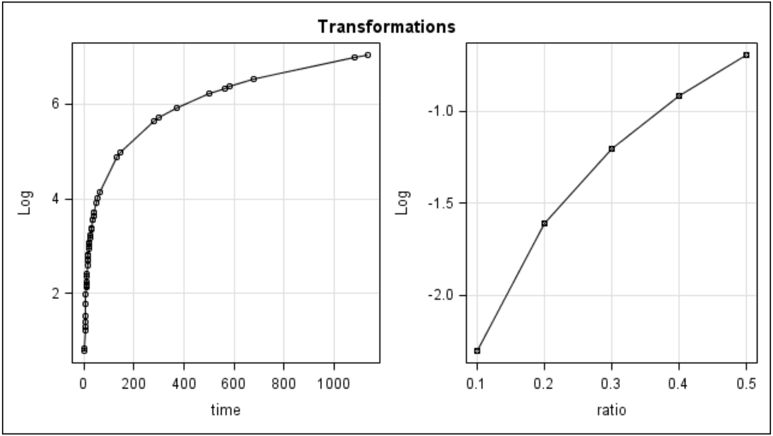


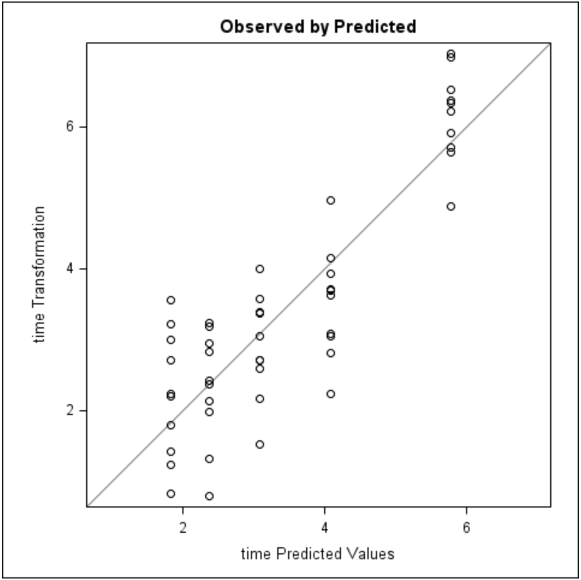

Observed by Predicted
time Transformation
600
400
200
0
0
200
400
600
time Predicted Values

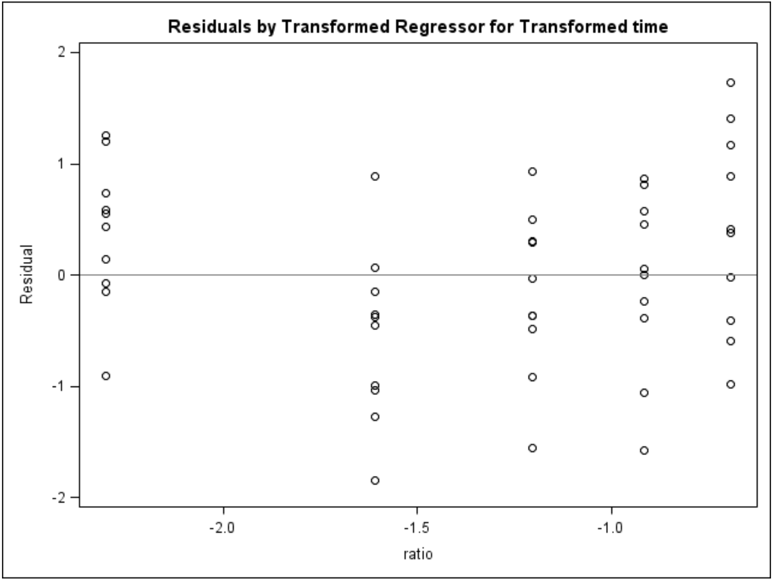

Residuals by Transformed Regressor for Transformed time
Residual
200
100
0
-100
-200
-300
0.0
0.1
0.2
0.3
0.4
ratio

What to do?

* The "best" fitted transformations look like log or square roots for both time and ratio.
* The log is also suggested by Box-Cox for time (not shown). Code: `model boxcox(time) = spline(ratio)`
* Refit the model with these simple functions:
* `model log(time) = log(ratio)`


Transformations
Log
time
0
200
400
600
800
1000
2
4
6
Log
ratio
0.1
0.2
0.3
0.4
0.5
-1.0
-1.5
-2.0


Observed by Predicted
time Transformation
time Predicted Values
2
4
6
2
4
6


Residuals by Transformed Regressor for Transformed time
Residual
ratio
2
1
0
-1
-2
-2.0
-1.5
-1.0

GAM in R

The package gam was written by Trevor Hastie (one of the inventors of GAM) and (in your instructor's opinion) is easier to use and gives nicer output that SAS PROC GAM.

Just as in PROC GAM, you tell the function gam which predictors to consider a transformation for and which to leave alone. Note that it does not make sense to consider a transformation of a categorical predictor.

The gam function gives plots of the full transformation $g_j(\cdot)$, not just the "wiggly" part $\tilde{g}_j(\cdot)$.

O-ring data

- Motivation: explosion of USA Space Shuttle Challenger on January 28, 1986.
- Rogers commission concluded that the Challenger accident was caused by gas leak through the 6 o-ring joints of the shuttle.
- Dalal, Fowlkes & Hoadley (1989) looked at number distressed o-rings (among 6) versus launch temperature (Temperture) and pressure (Pressure) for 23 previous shuttle flights, launched at temperatures between 53F and 81F.

O-ring variables

Data frame with 138 observations on the following 4 variables.

- ThermalDistress: a numeric vector indicating wether the o-ring experienced thermal distress.
- Temperature: a numeric vector giving the launch temperature (degrees F).
- Pressure: a numeric vector giving the leak-check pressure (psi).
- Flight: a numeric vector giving the temporal order of flight.

Dalal, S.R., Fowlkes, E.B., and Hoadley, B. (1989). Risk analysis of space shuttle : Pre-Challenger prediction of failure. *Journal of the American Statistical Association*, 84, 945-957.

Analysis in R

```
library(DPpackage); library(gam)
data(orings)
?orings
plot(orings)  # note that pressure only has three values!
fit=gam(ThermalDistress~s(Temperature)+Pressure+s(Flight),
 family=binomial(link=logit),data=orings)
par(mfrow=c(2,2))
plot(fit,se=TRUE)
summary(fit)
```

This fits the model

$$\text{logit}(\pi_i) = \beta_0 + \beta_1 T_i + \beta_2 P_i + \beta_3 F_i + \tilde{g}_1(T_i) + \tilde{g}_3(F_i).$$

More R examples

```
# example with linear log-odds
# parametric part significant, nonparametric part not significant
x=rnorm(1000,0,2); p=exp(x)/(1+exp(x)); y=rbinom(1000,1,p)
plot(x,y)
fit=gam(y~s(x),family=binomial(link=logit))
plot(fit,se=TRUE)
summary(fit)
fit$coef

# example with quadratic log-odds
# parametric part not significant, nonparametric part significant
p=exp(x^2)/(1+exp(x^2)); y=rbinom(1000,1,p)
plot(x,y)
fit=gam(y~s(x),family=binomial(link=logit))
plot(fit,se=TRUE)
summary(fit)
```